Amendments to the Claims:

This listing of claims will replace all prior versions, and listings, of claims in the application:

Listing of Claims:

1. (currently amended) A method ~~of~~ for implementing a cryptographic algorithm in an electronic device that includes performing modular multiplication of integers X and Y to produce a result R, where R=X.Y mod N, in a multiplication engine, the method comprising the steps of:

(a) fragmenting X into a first plurality of words x_n each having a first predetermined number of bits, k;

(b) fragmenting Y into a second plurality of words y_n each having a second predetermined number of bits, m;

(c) pre-calculating multiples of a word x_n of X in a pre-calculation circuit and using said pre-calculated multiples to derive products of the word x_n of X with each of the plurality of words y_n of Y;

(d) computing an intermediate result R_j as a cumulating sum derived from said ~~pre-calculated multiples~~ products;

(e) for each successive word of X, repeating the steps of pre-calculating and computing ~~so as~~ to generate successive intermediate results, R_j, for each of the first plurality of words x_n; ~~and~~

(f) providing as output each of the intermediate results R_j ~~so as~~ to form a final result~~.~~; and

(g) using the final result to complete an encryption or decryption operation within the electronic device.

2. (original) The method of claim 1 in which X is fragmented into n words of k bits each, according to the expression $X = x_{n-1}B_x^{n-1} + x_{n-2}B_x^{n-2} + \ldots + x_0$, where $B_x = 2^k$.

3. (original) The method of claim 1 in which Y is fragmented into n words of m bits each, according to the expression $Y = y_{n-1}B_x^{n-1} + y_{n-2}B_x^{n-2} + \ldots + y_0$, where $B_y = 2^m$.

4. (original) The method of claim 1 in which the step of computing an intermediate result R_j comprises generating a succession of terms x.y + c + z for addition, comprising the steps of:

(i) reading a pre-calculated multiple of a word x_n of X to form an $x_n.y_n$ product,

(ii) adding a carry word c_j, from a previous term;

(iii) adding a corresponding term, z, from a previous intermediate result;

(iv) fragmenting the result into a lower order m-bit word and a higher order, k-bit carry word;

(v) repeating steps (i) to (iv) for each of the $x_n.y_n$ products; and

(vi) after use of all $x_n.y_n$ products, forming a final term by adding the final carry word and corresponding term from the previous intermediate result.

5. (currently amended) The method of claim 4 wherein the step of computing the intermediate result is implemented as:

$R_j = x_{n-j+1}y_0 + (x_{n-j+1}y_1 + r_{j-1,0})B_y + (x_{n-j+1}y_2 + r_{j-1,1})\cancel{B_y}\underline{B_y}^2 + \ldots + (x_{n-j+1}y_{n-1} + r_{j-1,n-2})B_y^{n-1} + r_{j-1,n-1})B_y^n.$

6. (original) The method of claim 1 in which step (f) further includes combining all the intermediate results R_j to form R, according to the expression

$R = ((((x_{n-1}Y \bmod N)B_x + x_{n-2}Y) \bmod N)B_x + \ldots x_0Y) \bmod N.$

7. (original) The method of claim 4 in which step (i) comprises the steps of reading selected basic multiples of the word x_n of X and combining them to obtain the product $x_n.y_n$.

8. (previously presented) The method of claim 7 in which steps (i), (ii) and (iii) include combining the selected basic multiples the word of X, the carry word c_j, and the corresponding term z in an adder circuit.

9. (original) The method of claim 4 in which the corresponding term z from a previous intermediate result is the immediate less significant word from the previous intermediate result.

10. (original) The method of claim 4 in which the corresponding term z from a previous intermediate result is a (*k/m*)th less significant word from the previous intermediate result.

11. (original) The method of claim 1 in which the steps of pre-calculating comprise the steps of:

calculating pre-selected basic multiples of the word of X and

combining selected ones of the basic multiples to form a desired x.y product.

12. (original) The method of claim 4 in which the pre-calculation of multiples of a word of X takes place during step (vi) for the previous word.

13. (currently amended) Apparatus for performing modular multiplication of integers X and Y to produce a result R, where R=X.Y mod N, comprising:

means for fragmenting X into a first plurality of words x_n each having a first predetermined number of bits, k;

means for fragmenting Y into a second plurality of words y_n each having a second predetermined number of bits, m;

a pre-calculation circuit ~~(10)~~ for pre-calculating multiples of a word x_n of X and using said pre-calculated multiples to derive products of the word x_n of X with each of the plurality of words y_n of Y;

means for computing an intermediate result R_j as a cumulating sum derived from said ~~pre-calculated multiples~~ products; and

control means for controlling repetition of the pre-calculations and computing of

an intermediate result for each successive word of X ~~so as~~ to generate successive intermediate results, R_j, for each of the first plurality of words x_n~~,~~.

14. (previously presented) The apparatus of claim 13 in which the means for computing an intermediate result R_j generates a succession of terms x.y + c + z for addition, including:

(i) means for reading a pre-calculated multiple of a word x of X to form an x.y product,

(ii) means for adding a carry word c_j, from a previous term;

(iii) means for adding a corresponding term, z, from a previous intermediate result;

(iv) means for fragmenting the result into a lower order m-bit word and a higher order, k-bit carry word;

(v) control means for effecting repetition of the reading of a pre-calculated multiple and addition of the carry word and corresponding term for each of the x.y products and forming a final term by adding the final carry word and corresponding term from the previous intermediate result.

15. (canceled)

16. (canceled)

17. (canceled)

18. (canceled)

19. (canceled)

20. (new) The apparatus of claim 13, wherein the pre-calculation circuit comprises:

adder and shift circuits for deriving a plurality of basic multiples of x;

a plurality of registers for storing at least some of said plurality of basic multiples of x;

a plurality of multiplexers each receiving said basic multiples of x, each multiplexer having selection lines for receiving selected bits of a selected y word; and

a summation circuit for receiving the outputs from each multiplexer and combining them according to the numeric significance of the portion of the y word used as input to the respective multiplexer selection line.

21. (new) The apparatus of claim 20, wherein the plurality of registers correspond to selected odd basic multiples of x, even basic multiples of x being provided to each multiplexer by bit shifting lines coupled to selected ones of the plurality of registers.

22. (new) The apparatus of claim 20, wherein the plurality of multiplexers comprises a set of logic gates, each having a first input connected to receive a respective basic multiple of x, and a selection line to enable assertion of the basic multiple at an output thereof, wherein the summation circuit comprises a series of adders for receiving all asserted outputs of the series of logic gates, wherein only logic gates in the set of logic gates for which a selection input has changed will be switched during a change in the selected y word.

23. (new) The apparatus of claim 14 further comprising an adder circuit configured to combine the selected basic multiples of the word of X, the carry word c_j, and the corresponding term z.

24. (new) The apparatus of claim 14, wherein the corresponding term z from the previous intermediate result is the immediate less significant word from the previous intermediate result.

25. (new) The apparatus of claim 14, wherein the corresponding term z from the previous intermediate result is a (k/m)th less significant word from the previous intermediate result.